Detection and Prognostics on Low Dimensional Systems

Ashok N. Srivastava, *Member, IEEE*, Santanu Das

***Abstract*—This paper describes new algorithms for prognostics and anomaly detection on systems that can be described by low dimensional, potentially nonlinear dynamics. The methodology relies on estimating the conditional probability distribution of the output of the system at a future time given knowledge of the current state of the system. These conditional probabilities are estimated using a variety of techniques, including bagged neural networks and kernel methods such as Gaussian Process Regression (GPR) and compare the results against standard methods such as linear autoregressive models and the nearest neighbor algorithm. We demonstrate the algorithms on a real-world data set and a simulated data set. The real-world data set consists of the intensity of an NH_3 laser. The laser data set has been shown by other authors to exhibit low-dimensional chaos with significant drops in intensity. The simulated data set is generated from the Lorenz attractor and has completely known statistical characteristics. On these data sets, we show the evolution of the estimated conditional probability distribution, the way it can act as a prognostic signal, and its use as an early warning system. We also review a novel approach to perform Gaussian Process Regression with large numbers of data points.**

I. Introduction

This paper addresses the problem of making predictions of future events on systems that can be described by low-dimensional dynamical equations. We assume that we are given data from a data generating process that can be functionally described by the following equations:

$$\mathbf{h}_t = \Gamma(\mathbf{h}^*_{t-1}) \quad (1)$$

$$\mathbf{x}_t = \Psi(\mathbf{x}^*_{t-1}, \mathbf{h}^*_t, u_t) \quad (2)$$

$$y_t = \Omega(\mathbf{x}_t) \quad (3)$$

We assume that the function Γ determining the evolution of the hidden system state $\mathbf{h}_t$ is unknown. We also assume that the function Ψ, which generates the observed output of the system is also unknown. We assume that the vector $\mathbf{x}$ is an N dimensional state vector, and $\mathbf{x}^*_{t-1}$ is its history for the last d time steps: $\mathbf{x}^*_{t-1} = [\mathbf{x}_{t-d}, \mathbf{x}_{t-d+1}, ..., \mathbf{x}_{t-1}]$. The quantity u_t is the observed system input, and y_t is the observed scalar system output. We assume that the entire data that is available, covering both inputs and outputs is given by the set $(\mathcal{X}, \mathcal{Y})$.

The hidden state $\mathbf{h}_t$ is assumed to correspond to different mode configurations within the system. In the case where we assume that the hidden state takes on discrete values, $\mathbf{h}_t$ switches between M modes, each affecting the output dynamics Ψ. In the case of a failure of the system, $\mathbf{h}_t$ could move to a failed state, thus also changing the nature of the observed output. In other applications, $\mathbf{h}_t$ could be a continuous state variable, modeling a slow progression of the system from a normal state to a failed state [30]. Because we assume that we do not know the output function Ψ or the hidden state $\mathbf{h}_t$, we cannot rely on it to help us determine whether or not the observed y_t is anomalous. The problem that we address in this paper is to develop a method to discover whether or not the current observed value y_t represents an anomaly based on the observed history of the system.

Several approaches have been developed in the literature to address the problem of making future predictions on systems that can be described by Equations 2,3 and 1. Traditional approaches include those developed in the system identification community [16]. Other techniques include Hidden Markov Models, where the transitions between the M hidden discrete states are modeled as a first-order hidden Markov process [25]. The HMM allows for the dynamics of the system to be modeled but requires that a procedure (such as clustering or learning vector quantization) be used to develop a discrete representation of the system output. Other popular methods to convert the time series into symbolic representations include Piecewise Aggregate Approximation (PAA) [10]. Once the symbolic representation is generated it can be analyzed using the HMM. For many applications, the dwell time within a hidden state does not follow the exponential decay that arises from the standard HMM algorithm. Dong and He [6] [7] have recently developed this method for analyzing systems with hidden discrete transitions (as shown in Figure 1) using a hidden semi-Markov model (HSMM) where the dwell time within a state is modeled by a Gaussian distribution. Their work shows that the HSMM can lead to superior performance

on real-world applications compared with the standard HMM formulation.

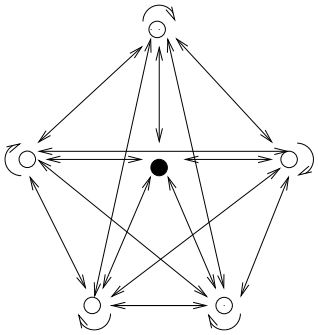

Fig. 1. This figure shows a finite state machine indicating a potential path for progression from normal operation (clear circles) to the failed state (solid circle). The model allows for the system to move from a failed state back to normal operation, which models intermittent problems that can arise in complex systems. For generality, we have included bi-directional arrows and a fully connected graph. In many cases, systems can only move into a failed state but not back to normal operation.

II. BACKGROUND

A time series is a collection of observations represented sequentially as a function of time. One of the major issues in time series analysis is to create mathematical models that predict the system behavior from a set of observations. The approach taken in the machine-learning community has been to create statistical models that learn a mapping from inputs to outputs. This does not require extensive knowledge of the physical system. In many cases, parametric models such as neural networks can produce high quality predictions [35]. The focus of the current study is restricted to the analysis of the time-series obtained from low dimensional nonlinear dynamical system that might exhibit chaotic behavior. The objective is to develop a prognostic model that can understand the dynamics of the system from the past observations and can forecast on how the system would evolve in the near future, within a certain prediction horizon. In many cases the decomposition of the time series into separate, simpler processes can be useful [31].

The simplest way to achieve this is to develop a model that learns from the training set in order to obtain a good estimate on the model parameters. Hence the first task is to adapt the knowledge about the model (to be developed) from the historical dataset. This is followed by obtaining the information on the next predicted (model) output conditioned over the current knowledge of the developed model. Once the single-step prediction is obtained, the main challenge lies in forming a multiple-step-ahead prediction scheme, irrespective of the model type. This is done by updating the model with the current estimate of the output at each iteration and repeating the predictor for k number of steps in the future.

The idea to predict future values as a linear combination of the preceding values was first introduced by Yule [39] as the auto-regressive process (AR). A comprehensive literature review on conventional techniques to select a model that could be used to forecast the behavior of the time has been provided by Weigend et al. [35]. In this literature, the authors provide some good insight on local-linear model, global AR, MA, and ARMA model and semi-local approach like Radial Basis Function (RBF) based model. K-nearest neighbor, a local average model, is a widely used supervised algorithm where the predicted instance is determined by the weighted average of the outputs, those present in the training set, with similar input vectors. Using different numbers of nearest neighbors, it is possible to find the optimal value of k for which the RMSE is minimized. McNames [22] presented a new method of optimizing the model parameters in order to minimize the multi-step cross validation error. In previous work [22], the author proposed the adoption of nearest trajectory model for time-series prediction. This method is motivated to search the closest trajectory points in the reconstructed state space as opposed to nearest neighbors. A comparative study on different nonparametric methods namely the nearest neighbors, RBF and nearest trajectory methods to predict chaotic time series can be found in [15]. The idea to obtain iterative time-series predictions using regression tree based approach has been addressed by Badel et al [1].

There is an increasing interest to investigate the appropriateness and adaptableness of global prediction methods to model nonlinear dynamical systems. Among all, neural networks is probably the most studied nonlinear function with input-output mapping capability. A standard way is to use MLP, RBF architecture in order to forecast time series with neural network. Several researchers have used hybrid predictors where the hybrid neural network is a conjunction of global approximation and local approximation.

Recently, probabilistic Gaussian Process models have gained a widespread popularity in the prognostic community. Given any initial value of the input, any of these models are intended to predict the distribution of future observation, based on its past knowledge. Naturally it is desirable to have a quantitative analysis as a measure of the uncertainty associated with each model prediction on future observations.

A detailed review on various sources of uncertainties has been well documented by Draper [9]. In the presence of model and input uncertainties, Gaussian Processes, a Bayesian based approach, is considered to be the most suitable technique where the predicted output can be shown to be a weighted sum of the training data and can be associated with an "error bar" which defines a confidence interval on the given prediction. Earlier in 1996, Neal [23] inferred that Gaussian Process (GP) models are equivalent to the neural networks with one hidden layer with infinitely large number of hidden neurons. Inspired by Neal's work, Rasmussen [26] first introduced the empirical formulation of Gaussian Processes in terms of probabilistic model using Bayesian treatment. He showed that simple matrix operations can be useful to develop a noise model and define the prior distributions over input variables.

Mackay [20] and Seeger [29] extended this research to show the associations of GP model to several other popular machine learning techniques like generalized Radial Basis Functions (RBF), Spline smoothing technique and Support Vector Machines. GP can be defined as a random function such that the output of the function, for a finite number of input points, constructs a random vector which is normally distributed. GP is fully specified by its mean function and covariance function. For zeros-mean process, the latter plays the prime role to characterize the process. It is interesting to note that a covariance function looks more like a kernel function that maps similar input data points to nearby neighborhood and the similarity is measured by the output of the function evaluated between any two input points. In statistical terms, the kernel function calculates the covariance between the outputs corresponding to different inputs and the choice of the covariance function typically depends on the prior assumption on the smoothness and continuity of the underlying function generated by a process. In other words the covariance function can be an automatic choice to characterize any process. Mathematically, a covariance function is termed valid if it produces a nonnegative definite covariance matrix for a given set of input points. Mackay [20] provided a detailed description on a wide variety of covariance functions.

III. Main Idea

The key idea discussed in this paper is to build a predictive model that estimates y_t given the history of past observations. Rather than creating a single 'point-estimate' of y_t, we estimate $P(y_t|\mathbf{y}^*_{t-1})$. In this formulation the mean of this quantity (obtained by computing the expected value) will produce an estimate for the future value of $\mathbf{x}_t$ while the variance of this distribution would quantify the uncertainty in the predictions. As that uncertainty changes in time, it can be indicative of an unanticipated change in the data generating process. This change could be due to several issues, including the movement of the hidden state $\mathbf{h}_t$ from one state to another.

For a true prognostic capability on low-dimensional systems, i.e., one where a forecast is made at a time horizon significantly far in the future compared to the natural frequency of the system, we need the ability to make long term predictions. Such predictions are theoretically impossible for chaotic systems [24] if the prediction horizon is on the same order as the Lyapunov time. This quantity is the amount of time that is required for a volume of phase space to expand to a size that completely covers the underlying dynamic attractor.

There are at least two ways to make predictions about an event in the future. One method is to make the prediction in such a way that the estimate $P(y_{t+\tau}|y^*_{t-1})$ for a fixed duration τ in the future. If τ is significantly longer than the natural period of the system as measured by the low frequency modes in the Fourier spectrum of the signal, such a model would make predictions that are fixed in time; it would only make predictions for those events that are exactly τ units in the future. The second method is to generate *iterated predictions* which rely on developing a model to estimate $P(y_t|y^*_{t-1})$ and then feeding the output of the model back into the input, thus producing an estimate of $P(y_{t+1}|\mu_t, y^*_{t-1})$, where μ_t is a statistic (such as the mean) computed from the distribution computed at time t [28], [33]. This form of iterated prediction is depicted in Figure 2.

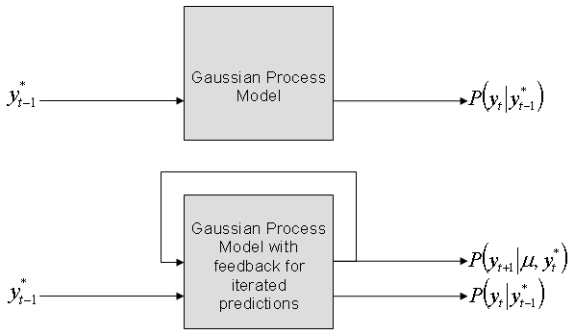

Fig. 2. The top panel in this figure shows a method for generating predictions of the quantity $P(y_t|\mathbf{y}^*_{t-1})$ using Gaussian Process regression. The lower panel shows a method to do iterated predictions, where a statistic such as the mean output of the model is fed back into the model to generate the next output.]

IV. ALGORITHMS

We provide a brief overview of three algorithms that we use for estimating $P(y_t|\mathbf{y}^*_{t-1})$. The first two algorithms, the k-nearest neighbor and the bagged neural network algorithms, have been developed for many applications and have been widely used in the literature [13]. These algorithms provide a benchmark for comparison against the performance of the Gaussian Process regression which has become popular in the machine learning community over the last ten years.

A. K-Nearest Neighbor

The k-nearest neighbor uses all available input data $\mathbf{X}$ and associated output data $\mathcal{Y}$ until time t to produce a prediction of y_{t+1}. Specifically, in order to estimate $P(y_t|\mathbf{y}^*_{t-1})$, given $\mathbf{x}^*_t$ we identify the k vectors in the data set that are closest to that vector in terms of the Euclidean distance. The mean of the outputs associated with those k vectors is used as an estimate of the expected value of the distribution, $E_P(P(y_t|\mathbf{y}^*_{t-1}))$ and the variance of the outputs is used to estimate $Var_P(P(y_t|\mathbf{y}^*_{t-1}))$. This method does not attempt to summarize the data in any way and simply uses a simple 'look-up' table to estimate the parameters of the distribution. These estimates are based on heuristic principles regarding the local distribution of data at time t.

B. Bagged Neural Networks

The bagged neural network [3] model consists of developing N feedforward multi-layer perceptrons using the available input and output data. Each model is made by taking a sample of data (with replacement) from the data set $(\mathcal{X}, \mathcal{Y})$. If each network is labeled as $G_i(\theta_i)$, the overall estimate of $E_P(P(y_t|\mathbf{y}^*_{t-1}))$ is given by:

$$E_P(P(y_t|\mathbf{y}^*_{t-1})) = \sum_{i=1}^{N} G_i(\mathcal{X}_i, \theta_i) \tag{4}$$

The estimate of the variance in the prediction, and thus the associated uncertainty is given by:

$$Var_P(P(y_t|\mathbf{y}^*_{t-1})) = \sum_{i=1}^{N} [G_i(\mathcal{X}_i, \theta_i) - E_P(P(y_t|\mathbf{y}^*_{t-1}))]^2 \tag{5}$$

It is important to note that this estimate of the uncertainty is a function of the heterogeneity of the data samples $\mathcal{X}_i$. In the event that all samples are identical, the only variation in the estimates will be due to variation in the initial starting point of the optimization procedure. This method is related to Ensemble Methods in machine learning and are widely used to estimate the mean and variance of the target distribution [34].

C. Gaussian Process Regression

A Gaussian Process is a stochastic process such that each finite subset of variables in the process is multivariate Gaussian distributed [27]. In 1996, Neal [23] noted that the if the weights and biases in a neural network are drawn from a Gaussian distributed, as the number of hidden units increases, the prior distribution over functions defined by such networks will converge to a Gaussian Process. This important result lead the machine learning community to research Gaussian Processes and Support Vector Machines.

Following the notation and derivation in Rassmussen and Williams [27], Gaussian Process Regression is a generalization of the standard linear regression model. We begin with a brief review of their Bayesian derivation of linear regression with the model $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ with additive Gaussian noise, we have:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{x}^T\mathbf{w} \\ y &= f(\mathbf{x}) + \epsilon_n \\ \epsilon_n &\sim N(0, \sigma_n^2) \end{aligned} \tag{6}$$

where $\mathbf{x}$ is the test vector and $\mathbf{w}$ is a set of weights. Assuming that we choose a prior distribution for the weights as Gaussian with zero mean and covariance matrix Σ_p, and that the noise is Gaussian and independent and identically distributed, with variance σ_n^2, we compute the posterior probability distribution of the weights given the data $(\mathcal{X}, \mathbf{y})$:

$$P(\mathbf{w}|\mathbf{X}, \mathbf{y}) = N(\frac{1}{\sigma_n^2} A^{-1}\mathbf{X}\mathbf{y}, A^{-1}) \tag{7}$$

where $A = \frac{1}{\sigma_n^2}\mathbf{X}\mathbf{X}^T + \Sigma_p^{-1}$. In order to make a prediction, given a test input $\tilde{\mathbf{x}}$, we compute the predictive distribution by averaging over the weights $\mathbf{w}$ and obtain:

$$P(f(\tilde{\mathbf{x}})|\tilde{\mathbf{x}}, \mathbf{X}, \mathbf{y}) = N(\frac{1}{\sigma_n^2}\tilde{\mathbf{x}}^T A^{-1}\mathbf{X}\mathbf{y}, \tilde{\mathbf{x}}^T A^{-1}\tilde{\mathbf{x}}) \tag{8}$$

where $A = \frac{1}{\sigma_n^2}\mathbf{X}\mathbf{X}^T + \Sigma_p^{-1}$. Using the so-called 'kernel trick', we obtain Gaussian Process Regression by assuming that we have a mapping $\Phi(\mathbf{x})$ that maps the original N dimensional data into a large, possibly infinite dimensional space. Replacing the independent variable $\mathbf{x}$ with its transformed version leads to the following posterior distribution:

$$P(f(\Phi(\tilde{\mathbf{x}}))|\tilde{(}\Phi\mathbf{x}), \mathbf{X}, \mathbf{y}) = N(\frac{1}{\sigma_n^2}\tilde{\Phi(\mathbf{x})}^T A^{-1}\mathbf{X}\mathbf{y}, \Phi(\tilde{\mathbf{x}})^T A^{-1}\Phi(\tilde{\mathbf{x}}) \quad (9)$$

This implies that the posterior distribution (eqn. 9) is also Gaussian, with the predicted mean $\hat{\mu}(\tilde{\mathbf{x}})$ and variance $\hat{\sigma}(\tilde{\mathbf{x}})$ for a given test input $(\tilde{\mathbf{x}})$. It is not very difficult to show that substituting the value of A and doing some simple matrix manipulations the predicted mean and variance,in the feature space, can be expressed as

$$\hat{\mu}(\tilde{\mathbf{x}}) = \Phi(\tilde{\mathbf{x}})^T\Sigma_p\Phi(\mathbf{x})^T[\sigma_n^2 I + \Phi(\mathbf{x})\Sigma_p\Phi(\mathbf{x})^T]^{-1}\mathbf{y} \quad (10)$$

$$\hat{\sigma}(\tilde{\mathbf{x}}) = \Phi(\tilde{\mathbf{x}})^T\Sigma_p\Phi(\tilde{\mathbf{x}}) - \Phi(\tilde{\mathbf{x}})^T\Sigma_p\Phi(\mathbf{x})[\sigma_n^2 I + \Phi(\mathbf{x})\Sigma_p\Phi(\mathbf{x})^T]^{-1}\Phi(\mathbf{x})\Sigma_p\Phi(\tilde{\mathbf{x}}) \quad (11)$$

It is important to note that the full Gaussian Process involves a two stage time complexity. The first one is definitely the time required to learn the hyperparameters, those used while constructing the covariance function. The immediate objective is to estimate the vector of the hyperparameters and the noise term by maximizing the log-likelihood. Throughout this research a stationary covariance function has been used and looks like:

$$C(x, x') = \theta_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{m}\frac{(x_i - x_i')^2}{\sigma_l^2}\right) + \theta_2 + \theta_3\delta_{ij} \quad (12)$$

where θ=[θ_1,θ_2,θ_3,σ_l] is the vector of hyperparameters of the covariance function. The parameters θ_1 and σ_l control the overall scale in vertical and horizontal variations respectively. Here θ_2 is the bias term and $\theta_3\delta_{ij}$ is the noise term where θ_3 is the variance of the noise and δ_{ij} is the Kronecker delta function. In order to train the Gaussian Process, these hyperparameters are optimized based on the seen examples. This means that the hyperparameters can be optimized with C_n^m combinations of seen examples, when *m* points are selected from a pool of *n* input points $(m < n)$. As m increases the computational complexity increases exponentially. A common way to compute the hyperparameter vector is to maximize the log-likelihood by taking the partial derivatives with respect to the hyperparameters. Further details of the cost function and optimization algorithm can be obtained in the following reference [20].

The second factor is the inversion of $(\sigma_n^2 I + \Phi(\mathbf{x})\Sigma_p\Phi(\mathbf{x})^T)$ which is a $n \times n$ matrix and compiling the matrix inversion subroutine can get extremely time and memory intensive with very large N. To handle this issue, we have adopted the Gaussian process method using V-formulation developed by Foster et. al [12]. In the proposed technique, the low rank approximation of larger matrices is being calculated using partial Cholesky factorization. In equation 10, the $n \times n$ matrix $(\Phi(\mathbf{x})\Sigma_p\Phi(\mathbf{x})^T)$ can be approximated by VV^T, where the $n \times m$ matrix V is constructed by partial Cholesky decomposition (PCD). These algebraic operations can be summarized as follows,

$$[\Phi(\mathbf{x})\Sigma_p\Phi(\mathbf{x})^T]_{n\times n} \overset{PCD}{\rightarrow} V_{n\times m}.V_{m\times n} \quad (13)$$

$$[\Phi(\tilde{\mathbf{x}})^T\Sigma_p\Phi(\mathbf{x})^T]_{p\times n} \overset{PCD}{\rightarrow} V^*{}_{p\times m}.V_{m\times n} \quad (14)$$

Equation 10 can be rewritten as,

$$\hat{\mu}(\tilde{\mathbf{x}}) = V^*V^T[\sigma_n^2 I + VV^T]^{-1}\mathbf{y} \quad (15)$$

Lemma 1: $V^T[\sigma_n^2 I + VV^T]^{-1} = [\sigma_n^2 I + V^TV]^{-1}V^T$

The V formulation takes advantage of the above lemma to reconstruct both equation 10 and 11. The details on the proof can be obtain in the following literature [12]. Using Lemma 1, one can rewrite equation 15 as,

$$\hat{\mu}(\tilde{\mathbf{x}}) = V^*[\sigma_n^2 I + V^TV]^{-1}V^T\mathbf{y} \quad (16)$$

Equation 16 is the basis of V formulation. Instead of directly inverting the $n \times n$ matrix $[\sigma_n^2 I + VV^T]$, the algorithm address the inversion of $n \times m$ matrix $[\sigma_n^2 I + V^TV]$, uses the partial Cholesky factorization. For smaller *m*, the regression algorithm with V formulation is much faster and also numerically stable.

V. Lorenz Dynamics and Laser Data

In this section we provide a small description on the Far-infrared (FIR) laser system,the prime source of our experimental dataset, used in this research to test the machine learning algorithms. It is very complicated to develop an accurate physical model of the FIR laser system. However several researchers have shown the connection of the experimentally measured electric field from NH_3 laser system to that of the dynamics of Lorenz's model [36]. The Lorenz's equation, as proposed by Hankel [14], can approximate the dynamical behavior of the optically pumped NH_3 single mode laser field. The equations for the Lorenz model can be expressed as follows,

Input: X (Input), t (target), C (Covariance Function) R_{max} (maximum rank), $\tilde{\mathbf{x}}$ (test input) and N_{iter} (Number of iterations).
Step 1: Randomly select a subset m of n input points to train hyperparameters (where $m < n$).
Step 2: Optimize hyperparameter vector θ.
a) Initialize θ=[θ_1,θ_2,θ_3,σ_l]
b) Compute covariance matrix C using n input points.
c) Compute low-rank approximation of covariance matrix constructed in (b).
d) Construct log-arithmetic likelihood function.
e) Maximize cost function with respect to each hyperparameter.
f) Obtain the optimized hyperparameter values.
Step 3: Construct Model.
g) Redefine low-rank approximation of covariance matrix using optimized hyperparameters.
Step 4: Make Predictions
for k := 1 to N_{iter}
$\tilde{\mathbf{x}}$ = $\tilde{\mathbf{x}}$;
h) Compute covariance matrix between test points and active set.
i) Redefine low-rank approximation of covariance matrix constructed in (g).
j) Compute and Store k^{th} predictive mean and variance.
k) Update $\tilde{\mathbf{x}}$ with k^{th} predicted mean.
end
Output: $\hat{\mu}(\tilde{\mathbf{x}})$ (Predicted mean), $\hat{\sigma}(\tilde{\mathbf{x}})$ (variance)

Fig. 3. Steps of Iterative Gaussian Process Algorithm.

$$\dot{u} = -\sigma(u - v) \tag{17}$$

$$\dot{v} = -u(w - r) - v \tag{18}$$

$$\dot{w} = uv - bw \tag{19}$$

The set of differential equations 19 describes a nonlinear dynamical system and for certain values of σ, b and r, the evaluation of the state vectors (u, v, w) give rise to the famous Lorenz-like spiral chaos, known as Lorenz attractor. Figure(4) shows a typical three dimensional representation of the Lorenz attractor, numerically simulated using 4^{th} order adaptive Runge-kutta integration scheme. The state variables and the control parameters in the Lorenz equation are closely related with the physical quantities of the laser physics. The details of these relationships and their interpretations can be found in the following literatures [14]. The absolute square of the state variable (u) at each time instant is

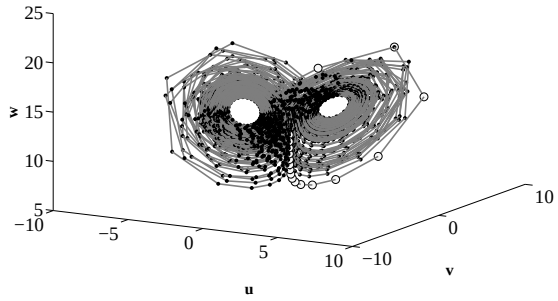

Fig. 4. This figure shows a three dimensional representation of the Lorenz attractor evolved with time. The integration time step is 0.05 second with $\sigma = 2$, $b = 0.25$ and $r = 15$. Here u, v, w are the state variables.

analogous to the electric field intensity field of the NH_3 laser system. The experimental dataset on the intensity pulsation (figure 5), sampled at 12.5 MHz, has 25000 8-bit sample points obtained from an optically pumped 81.5-micron NH_3 FIR laser. The data has a signal-to-noise ratio of 300. Further details of the measurement setup can be found in the literature by Huebner et al [14]. and the dataset has been obtained from the Santa Fe Time series competition. The Lorenz-like chaotic patterns observed in the intensity pulsation of the laser can be theoretically generated using Lorenz model with the control parameters adjusted to $\sigma = 2$, $b = 0.25$ and $r = 15$.

VI. Applications to Mechanical Systems

Built-in diagnostics and prognostics schemes, to interrogate the condition of components or sub-components of a system, has tremendous potential as this could offer cost effective solution to improve performance and reliability of these aging or newly designed complex systems. Recent developments in systems health monitoring indicate that fault detection systems make use of either passive type or active sensing devices [5], [32], [37], [38], those connected in a discrete or continuous fashion. In most cases, these sensing devices monitors one or more state variables depending on the nature of the application and the outputs are typically a series of signatures, a measure of different system states. In structural applications, failure generally initiates in regions with high stress concentration which leads to complex crack patterns within that material.Under the influence of sufficiently large external energy, the nucleated crack might propagate until complete failure occurs. These progressive damage could cause a change in the

dynamic characteristics and also reduce the strength of the structure, depending on the geometry and location of the fault. The dynamic response of the system is even more complex due to complicated stress wave patterns, material nonlinearity and the inherent nonlinearity associated with continuously evolving geometry.

Recent studies have revealed that techniques based on the nonlinear manifestation of defects through generation of multiple harmonics are very promising for nondestructive inspection. The history of nonlinear techniques for the NDE has been mentioned by Donskoy and Bovsunovskii. Donskoy et al. [8] addressed the vibromodulation technique that has been implemented in the N-Scan damage detection system. This technique utilized the effect of the nonlinear interaction of ultrasonic waves and vibrations at the interface of the defect. Bovsunovskii et al. [2], [21]investigated the resonance, sub-, and super-resonance vibrations of an elastic body with a crack, and presented analytical, numerical, and experimental results of a cracked body with one degree of freedom. A simple analytical model has been presented by Yamanaka et al. [17]–[19] to explain the interaction between crack planes using van der Waals interatomic force. This approach addressed the fundamental problem regarding the detection of a closed crack in ultrasonic testing. The paper proposes a new detection technique based on the subharmonic components which is generated due to the nonlinear interaction of crack surface and forcing function. However, he showed that under certain parametric condition, the vibration signals, that represent the crack opening displacement, can exhibit chaotic behavior. Foong et al. [11] have shown that the response of nonlinear dynamical systems can have chaotic oscillation under fatigue crack growth.In vibration based or wave based approach, the outputs of sensors are basically time series and analysis of these signals are far more complicated than anticipated, for real life heterogeneous systems. This circumstances gives rise to an increasing interest to explore different time-series analysis on nonlinear chaotic systems.

VII. Result Section

Results are presented to demonstrate the applicability of the iterative Gaussian process method to predict the futuristic behavior of the experimental data, obtained from NH_3 laser system. In the first set of analysis, part of the data (1200 sample points) has been assigned for training purpose as shown in figure 5. Three different data segments, each of 500 sample points in length, have been arbitrarily chosen from the rest of the time series and have been assigned for test purpose. Once the

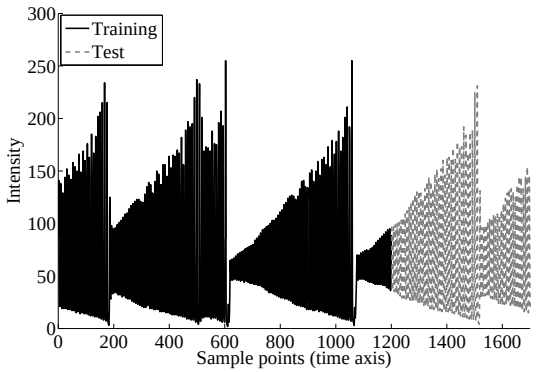

Fig. 5. This figure represents the time history of the intensity experimentally generated from NH_3 laser system. The first 1200 sample points of the data (dark) has been used for training and the rest 500 points (light) for test purpose.

model is appropriately trained, it is possible to iteratively predict q steps ahead in time for each test case, given the very first sample point of the corresponding test case. It should be noted that, for a chaotic time series, the forecast of q steps in future is restricted to certain prediction horizon that can be calculated from Lyapunov's exponent, if known. However to predict within this limit, it is necessary to train the model such that it learns on the underlying dynamics of the system from the scalar observations and the extraction of the dynamics can be achieved by utilizing the time embedding technique. According to Taken's theorem [4], given a finite set of scalar observations, it is possible to reconstruct the attractor in the phase space with an appropriate choice of time delay (τ) and embedding dimension (D). Given a time series $x(t)$ of length N, the delay vectors $y(t)$ with reassigned length of $N-(D-1)\tau$ data points can be represented as follows,

$$y(t) = [x(t)x(t+\tau)2x(t+2\tau)...x(t+(D-1)\tau)]^T \quad (20)$$

[**need some feedback on how much to explain on this topic. Some part of the results have been illustrated in the conclusion section**]

VIII. Conclusions

The present study describes a methodology using Gaussian process to study low dimensional dynamical chaotic systems. Prognostic algorithms are aimed not only to forecast statements on the upcoming future events but also to generate warning signals,ahead in time, regarding any possible catastrophic failures if anticipated with confidence. In this work, it has been

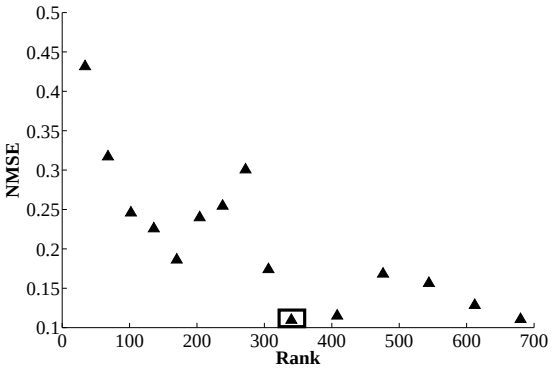

Fig. 6. The normalized mean square error (NMSE) of prediction has been plotted for different values of rank (set to maximum rank) that has been used to reconstruct the low rank approximation of the matrix using partial Cholesky factorization. The maximum rank is set to 340 which is $10\times$ dimension of the training set.

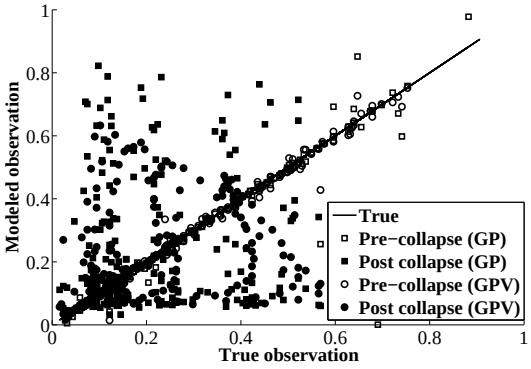

Fig. 7. The above figure represents the scatter plot of the true observation and the modeled data. The straight line represents the ideal case when the true observation has been plotted across itself. The hollow and solid circles represent the pre-collapse and post collapse points (respectively), modeled using GP with V-formulation. Similarly hollow and solid squares correspond to pre-collapse and post collapse points (respectively), modeled using GP without V-formulation. Note that this is the same test case illustrated in figure 5.

TABLE I

THE TABLE REPRESENTS A QUALITATIVE COMPARISON OF NORMALIZED MEAN SQUARE ERROR (NMSE) USING DIFFERENT MODELS. THE RESULTS ARE BASED ON 450 SAMPLE POINTS OF TRUE AND MODELED OBSERVATIONS. THE FIRST 1200 SAMPLE POINTS SERVES AS TRAINING SET AND PREDICTIONS HAVE BEEN DONE ON THREE SEPARATE TEST CASES (A, B, C). NOTE FOR K-NN METHOD, THE VALUE OF K IS SET TO 30.

Test	GP	GP-V	B NET	K-NN
A (1201-1650)	0.1594	0.1095	0.2390	0.2574
B (2001-2450)	0.1820	0.0873	0.2862	0.2495
C (3201-3650)	0.1005	0.1246	0.1913	0.1985

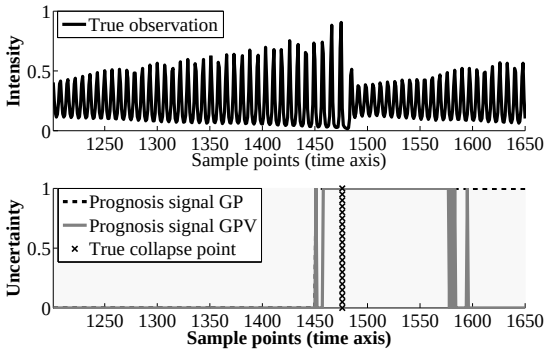

Fig. 8. The top image in this plot shows the true observation (test vector) that has been predicted. The lower half of the figure demonstrates the prognostic signal, a digitalized indicator,that can be generated prior to the true collapse. Both GP with and without V-formulation is able to flag the forthcoming collapse event. Note that this is the same test case illustrated in figure 5.

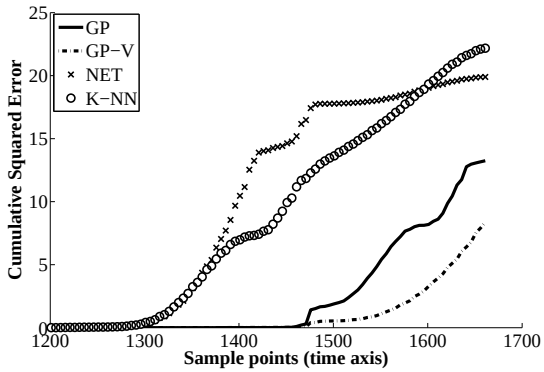

Fig. 9. This figure represents the comparison based on cumulative prediction errors resulting from different models. Clearly it shows that both Gaussian process with and without V-formulation accumulates less prediction error within a certain horizon, compared to neural network and K nearest neighborhood with $k = 30$. Note that this is the same test case illustrated in figure 5.

TABLE II

THE TABLE BELOW REPRESENTS THE PREDICTION HORIZON, RELATIVE TO THE START POINT, FOR DIFFERENT MODELS (FOR CLARITY, WE INDICATE THAT THE PREDICTED HORIZON CORRESPONDS TO THE SAMPLE POINT FOR WHICH CUMULATIVE SQUARE ERROR IS ≤ 1). NOTE FOR K-NN METHOD, THE VALUE OF K IS SET TO 30.

Test	GP	GP-V	B NET	K-NN
A (1201-1650)	274	346	122	119
B (2001-2450)	184	209	144	115
C (3201-3650)	312	313	235	160

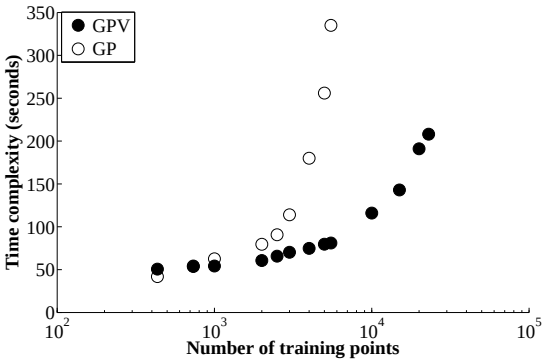

Fig. 10. This figure addresses the computational issues associated with Gaussian process with and without V-formulation. It can be seen that the GP with V-formulation has a superior capability in handling both time and space complexities, particularly with larger training points as shown in log scale.

TABLE III
THE VALUES IN THE TABLE BELOW INDICATES THE NUMBER OF SAMPLE POINTS BY WHICH THE PROGNOSIS SIGNAL LEADS THE TRUE COLLAPSE POINT. THE PROGNOSIS SIGNAL IS SET TO 1 ONCE THE UNCERTAINTY ASSOCIATED WITH EACH PREDICTION CROSSES A PREDEFINED THRESHOLD.

Trail	GP	GPV
A (1201-1650)	24	24
B (2001-2450)	30	8
C (3201-3650)	34	42

demonstrated how Gaussian Process can be trained with time embedded historical dataset to make short term predictions on future evolutions of the laser system. In Fig. 8, the intensity plot shows that the developed procedure can accurately predict the rapid oscillations that grows monotonically with time. It has been observed that the generated forecast holds good correlation with the original data until the collapse point is reached. In the post-collapse region the model starts accumulating error and looses it's prediction capability. Hence it can be concluded that Gaussian Process as a predictor can provide adequate feedback on the event of a collapse.Also a comparative study has been conducted on the performance of the Gaussian Process as a predictor and the results indicate that Gaussian Process emerges with a better score when compared with bagged Neural Network and K-nearest Neighbor based algorithms.

As mentioned earlier that in Gaussian Process, the predicted output is associated with an "error bar" that describes the confidence limit on that prediction. In this research, one of the key intensions is to generate a prognosis signal that can be used to forecast and map the future collapse event. This article has demonstrated a state-of-art that can generate a warning signal in the pre-collapse stage such that precautionary measures can be taken before the collapse occurs. The idea is based on the fact that, given a trained process, the uncertainties associated with the prediction of a future (system) collapse would intuitively be greater than predicting alike collapse example from the subset of the historical data. Once the uncertainty associated with the forecast crosses certain threshold, the output is set to "high"' status, indicating a probable collapse might occur. Fig. 8 shows such a digitalized output and for different test cases,it has been observed that the warning signal is set to "high"' much before the actual collapse occurs, as shown in Table III. Hence there is an immense potential of using such warning signals for prognosis purpose on real life systems, specially where the historical data has some of the examples with the associated collapse patterns.

REFERENCES

[1] L. Mercier A. E .Badel, D.Guegan and O. Michel, *Comparison of several methods to predict chaotic time series*, Acoustics, Speech, and Signal Processing, IEEE.

[2] A. P. Bovsunovskii, *Vibrations of a nonlinear mechanical system simulating a cracked body*, Strength of Materials (July).

[3] Leo Breiman, *Bagging predictors*, Machine Learning **24** (1996), no. 2, 123–140.

[4] T. Buzug and G. Pfister, *Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors*, Phys. Rev. A **45** (1992), 7073.

[5] F. K. Chang, *Built-in damage diagnostics for composite structures*, August.

[6] M. Dong and D. He, *Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis*, European Journal of Operational Research **178** (2007), 858–878.

[7] _____, *A segmental hidden semi-markov model based diagnostics and prognostics framework and methodology*, Mechanical Systems and Signal Processing **21** (2007), 2248–2266.

[8] D. Donskoy, *N-scan: New vibromodulation system for detection and monitoring of cracks and other contact-type defects*, Smart Structures and Materials (August).

[9] D. Draper, *Assessment and propagation of model uncertainty*, Journal of the Royal Statistical Society, Series B (Methodological) **57** (1995), 45.

[10] M. Pazzani E. Keogh, K. Chakrabarti and S. Mehrotra, *"dimensionality reduction for fast similarity search in large time series databases"*, Knowledge and Information Systems **3** (2001), no. 3, 263–286.

[11] C. H. Foong, E. Pavlovskaia, and and W. F.Deans M. Wiercigroch, *Chaos caused by fatigue crack growth*, Chaos, Solitons and Fractals **16** (2003), 651.

[12] L. Foster, A. Waagen, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, P. Gazis, A. N. Srivastava, and M. Way, *Improved linear algebra methods for redshift computation from limited spectrum data* -II, (2007).

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, Springer, 2001.

[14] U. Huebner, C. O. Weiss, N. B. Abraham, and D. Tang, *Lorenz-like chaos in nh3-fir lasers (data set a)*, Time Series Prediction:Forecasting the Future and Understanding the Past (N. Gershenfeld A. Weigend, ed.), 1994.

[15] J.D.Wichard and M. Ogorzalek, *Iterated time series prediction with ensemble models*, Proceedings of the 23rd IASTED International Conference on Modeling, Identification and Control, 2004.
[16] J. Juang, *"applied system identification"*, "Prentice-Hall", 1994.
[17] T. Mihara K. Yamanaka and T. Toshihiro, *Evaluation of closed cracks by analysis of subharmonic ultrasound*, Non-Destructive Testing and Condition Monitoring (November).
[18] T. Mihara K. Yamanaka and T. Tsuji, *Evaluation of closed cracks by model analysis of subharmonic ultrasound*, Japanese journal of applied physics (September).
[19] T. Mihara M. Akino and K. Yamanaka, *Fatigue crack closure analysis using nonlinear ultrasound*, February.
[20] D.J.C. MacKay, *Gaussian processes - a replacement for supervised neural networks?*, Advances in Neural Information Processing Systems **9** (1997).
[21] V. V. Matveev and A. P. Bovsunovskii, *Some aspects of vibration of an elastic body with a "breathing*.
[22] J. McNames, *Local averaging optimization for chaotic time series prediction*, Neurocomputing.
[23] Radford M. Neal, *Bayesian learning for neural networks*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
[24] R. Mainieri G. Tanner G. Vattay N. Whelan P. Cvitanovic, R. Artuso and A. Wirzba, *Chaos: Classical and quantum*, http://ChaosBook.org, 2007.
[25] L. R. Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE **77** (1989), no. 2, 257–286.
[26] C. E. Rasmussen, *Evaluation of gaussian processes and other methods for non-linear regression*, 1996, Ph.D. thesis,Department of Computer Science, University of Toronto.
[27] C. E. Rasmussen and C.K.I Williams, *"gaussian processes for machine learning"*, MIT Press, 2006.
[28] T. Sauer, *Time series prediction by using delay coordinate embedding*, Time Series Prediction: Forecasting the Future and Understanding the Past (N. Gershenfeld A. Weigend, ed.), 1994.
[29] M. Seeger, *Gaussian processes for machine learning*, International Journal of Neural Systems **14** (2004), 1.
[30] A. N. Srivastava, *Discovering system health anomalies using data mining techniques*, Proceedings of the 2005 Joint Army Navy NASA Airforce Conference on Propulsion (2005).
[31] A. N. Srivastava, R. Su, and A. Weigend, *Data mining for features using scale sensitive gated experts*, IEEE Transactions on Pattern Analysis and Machine Intelligence (1999).
[32] I. A. Viktorov, Rayleigh and Lamb Waves: Physical Theory and Applications, Plenum Press, New York, 1967.
[33] E. Wan, *Time series prediction by using a connectionist network with internal delay lines*, Time Series Prediction: Forecasting the Future and Understanding the Past (N. Gershenfeld A. Weigend, ed.), 1994.
[34] M. J. Way and A. N. Srivastava, *Novel methods for predicting photometric redshifts from broad band photometry using virtual sensors*, Submitted to Astrophysical Journal (2006).
[35] A. Weigend and N. Gershenfeld, *Time series prediction: Forecasting the future and understanding the past*, Addison-Wesley, 1994.
[36] C. O. Weiss, W. Klische, N. B. Abraham, and U. Hbner, *Comparison of nh3 laser dynamics with the extended lorenz model*, Applied Physics B **49** (1989), 211.
[37] M. Wenger, P. Blanas, R. J. Shuford, and D. K. Das-Gupta, *Acoustic emission signal detection by ceramic/polymer composite piezoelectrets embedded in glass-epoxy laminates*, Polymer Engineering and Science **36** (1996), 2945.
[38] D. C. Worlton, *Ultrasonic testing with lamb waves*, Nondestructive Testing **15** (1957), 218.
[39] G. U. Yule, *On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers*, Philosophical Transactions of the Royal Society of London **226** (1927), 267.

ACKNOWLEDGMENT

This research was supported in parts by: IVHM project, NASA Ames Research Center, Nasa Aeronautics Research Mission Directorate, Aviation Safety Program and NESC grant, a collaborative research between Arizona State University and NASA Ames Research Center, grant number NNA07CN64A, technical monitor Dr. Rodney Martin. The authors would like to thank Dr. Leslie Foster and Dr. Paul Gazis for providing important feedbacks while conducting this research.